



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

09/750,475

12/28/2000

Lynh Nguyen

STL919990134US3/A8644

7832

46159

7590

01/30/2013

SUGHRUE MION PLLC

USPTO CUSTOMER NO WITH IBM/SVL

2100 PENNSYLVANIA AVENUE, N.W.

WASHINGTON, DC 20037

EXAMINER

NGUYEN, THU V

ART UNIT

PAPER NUMBER

2452

MAIL DATE

DELIVERY MODE

01/30/2013

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte LYNH NGUYEN

Appeal 2013-002018
Application 09/750,475¹
Technology Center 2400

Before JOSEPH F. RUGGIERO, CAROLYN D. THOMAS, and
STANLEY M. WEINBERG, *Administrative Patent Judges*.

THOMAS, *Administrative Patent Judge*.

DECISION ON APPEAL

¹ The real party in interest is International Business Machines Corporation.

STATEMENT OF THE CASE

Appellant seeks our review under 35 U.S.C. § 134 of the Examiner's final decision rejecting claims 1, 6-8, 13-15, 18-22, and 24, which are all the claims remaining in the application. Claims 2-5, 9-12, 16, 17, and 23 are cancelled. We have jurisdiction over the appeal under 35 U.S.C. § 6(b).

We AFFIRM.

The present invention relates generally to automatically re-establishing a connection to a data source accessible by a plurality of remote applications. *See* Spec. 1, ll. 13-14.

Claim 1 is illustrative:

1. A method for automatically re-establishing a connection to a data source accessible by a plurality of remote applications, the method comprising:

providing at least one interface module configured to interface with a remote application;

providing at least one port module configured to interface between the interface module and the data source;

providing a connection manager to facilitate the interface between the interface module and the at least one port module;

detecting unavailability of the data source, by the at least one port module, in response to an initial request for the data source by the remote application;

dynamically detecting availability of the data source, by the at least one port module, in response to a subsequent request for the data source; and

re-connecting the data source to the remote application in response to the subsequent request,

wherein the at least one port module sends an error message to the interface module indicating the unavailability of the data source,

reestablishes a connection with the data source, and reconnects the remote application to the data source directly communicating with the remote application, and

wherein the at least one port module bypasses the connection manager in the subsequent request.

Appellant appeals the following rejection:

Claims 1, 6-8, 13-15, 18-22, and 24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Yousefi'zadeh (US 6,950,848 B1, Sept. 27, 2005), Guenthner (US 6,134,588, Oct. 17, 2000), and Albert (US 6,549,516 B1, Apr. 15, 2003).

ANALYSIS

Appellant argues claims 1, 6-8, 13-15, 18-22, and 24 as a group (App. Br. 6-10). For claims 6-8, 13-15, 18-22, and 24, Appellant repeats the same argument made for claim 1. We will, therefore, treat claims 6-8, 13-15, 18-22, and 24 as standing or falling with claim 1. *See* 37 C.F.R. § 41.37(c)(1)(vii). *See also In re Young*, 927 F.2d 588, 590 (Fed. Cir. 1991).

Issue 1: Did the Examiner err in finding that Albert discloses wherein the at least one port module bypasses the connection manager in the subsequent request, as claimed in claim 1?

Appellant contends that “future packets are not subsequent requests for the data source. Accordingly, Albert does not teach or suggest ‘wherein the at least one port module bypasses the connection manager in the subsequent request.’” (App. Br. 8.)

The Examiner found that “it is reasonable to interpret each of *Albert*’s subsequent packets within a data flow as a ‘subsequent request.’ There is no language in the claim that prohibits interpreting ‘request’ as a ‘packet.’” (Ans. 9.) The Examiner further concluded that “[b]ecause the future packets are directed to the data source, it is reasonable to interpret the packets as being requests [sic] for the data source” (*id.* at 10).

In essence, the Examiner admits that both *Yousefi ’zadeh* and *Guenthner* do not explicitly disclose *wherein the at least one port module bypasses the connection manager in the subsequent request*, but instead relies upon *Albert* to disclose such features. As such, we shall look for error in the Examiner’s interpretation of *Albert*.

Specifically, *Albert* discloses that “[f]uture packets in either flow sent from the client or the host match the affinity key in one of the fixed affinities and are handled by the forwarding agent . . . It is no longer necessary to forward such packets to the service manager” (*Albert*, col. 13, ll. 4-8). In other words, *Albert* discloses that future packets are handled by the forwarding agent and not necessarily the service manager. Appellant disputes that such “future packets” are equivalent to the claimed “subsequent requests for the data source.” However, we find that the Examiner associates *Albert*’s “future packets” with “future requests” (i.e., subsequent requests) for the data source, seemingly because the packets in *Albert* are *not* shown to be automatically transmitted, i.e., without first being requested. At least for this reason, we find the Examiner’s interpretation of *Albert*’s packet flow (e.g., being the result of a request) reasonable.²

² In our previous Decision issued December 4, 2009, we further found that *Guenthner* discloses detecting unavailability (e.g., a “Bad” server) and

Thus, based on the record before us, we find no error in the Examiner's interpretation of *Albert*, essentially for the reasons indicated by the Examiner.

Issue 2: Did the Examiner err in finding that Guenthner discloses sending an error message to the interface module indicating the unavailability of the data source, as claimed in claim 1?

Here, the Examiner relies upon *Guenthner* to teach the above noted feature (*see* Ans. 6 and 10). As such, we shall look for error in the Examiner's interpretation of *Guenthner*.

Appellant contends that "Guenthner merely discloses that if no entry is 'Current', then the routine returns an error. However, there is no teaching or suggestion of (i) an error message being sent to the interface module and (ii) [] error message indicating the unavailability of the data source" (App. Br. 10).

The Examiner found that Guenthner discloses "providing an error indication to the user" and that "[t]he error is provided to the user when there are no data sources available based on a determination from a list of servers" (Ans. 10). We agree with the Examiner.

Specifically, Guenthner discloses that "[w]hen the browser issues a request, a name service returns a list of IP addresses that may service that request. . . . and IP addresses are selected from the list. . . until a connection to an appropriate server is obtained" (Abstract). Guenthner further discloses

dynamically detecting (e.g., at frequent intervals) availability of the server in response to a subsequent request (e.g., retrying entries)(Decision, 6).

that “[i]f a particular server fails to respond in response to a selected IP address, a ‘timeout’ policy is preferably enforced. In particular, the browser marks (in the HAL) the failed entry ‘Bad’ for a given time period” (col. 5, ll. 57-60). In Guenthner, the system “returns an IP address list **52** for use by the browser, or it returns an error. If the Renew HAL routine returns an error, the routine branches to step **82** and provides an error indication to the user” (col. 6, ll. 37-40). In addition, “[i]f no entry is ‘Current’, the routine branches to step **112** and returns an error” (col. 7, ll. 50-51).

In other words, Guenthner discloses a method for securing a list of IP addresses of Web servers that host content requested by a client. In Guenthner, if a server fails (i.e., is unavailable) an error indication is returned to the user (i.e., user interface). The recited “sends an error message to the interface module indicating the unavailability of the data source” is strikingly similar (at least conceptually) to Guenthner’s teachings noted *supra*, and the Examiner’s reliance on this functionality is therefore persuasive as Guenthner teaches sending an error indicator to an interface.

Therefore, on this record, we sustain the Examiner’s obviousness rejection of independent claim 1 for essentially the same reasons argued by the Examiner, as discussed above. For the reasons previously discussed regarding claim 1, we also sustain the Examiner’s § 103 rejection of claim 6-8, 13-15, 18-22, and 24, as these claims are not separately argued.

DECISION

We affirm the Examiner’s § 103 rejection of claims 1, 6-8, 13-15, 18-22, and 24.

Appeal 2013-002018
Application 09/750,475

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a). *See* 37 C.F.R. § 1.136(a)(1)(iv) .

AFFIRMED

kis